

The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Comp 411 Computer Organization

Fall 2009

Problem Set #1

Issued Wed. 9/02/09; Due Wed. 9/09/09 (beginning of class)

**Homework Information:** Some of the problems are probably too long to be done the night before the due date, so plan accordingly. Late homework will not be accepted. Feel free to get help from others, but the work you hand in should be your own. You may **not** use solutions to a previous year's homework to aid you in completing this assignment. Please **enter your answers in the space provided**, but feel free to attach any additional sheets of scratch work.

**Problem 1: Information Encoding (18 points)**

In lecture we learned that information resolves uncertainty, and that information is measured in units of *bits*. In order to uniquely identify one of  $N$  equally likely alternatives,  $\log_2 N$  bits of information must be communicated. Use this information to solve the problems below:

- a) How many bits are necessary to encode an integer in the range of 0 to 15 (inclusive)?

Answer:

- b) How many bits are necessary to encode an integer in the range of 0 to 256 (inclusive)?

Answer:

- c) How many bits are necessary to encode an integer in the range of -256 to 255 (inclusive)?

Answer:

- d) Consider a standard card deck (4 distinct suits; values 2-10, J, Q, K, A). How many bits are necessary to encode a card's suit?

Answer:

- e) How many bits are necessary to encode a card's value?

Answer:

- f) Suppose we create a deck of cards that consists only of the number cards (2-10) and only has three suits. What is the smallest number of bits necessary to uniquely encode the new deck?

Answer:

**Problem 2: Two's Complement Representation (24 points)**

Most computers choose a particular *word length* (measured in bits) for representing integers and provide hardware that performs operations on word-size operands. Many current generation processors have word lengths of 64 bits. Restricting the size of the operands and the result to a single word means that the arithmetic operations are actually performing arithmetic modulo  $2^{64}$ .

- a) How many different values can be encoded in a 64-bit word? Give your answer in terms of a power of 2. Answer:

- b) If the 64-bit word must allow negative numbers, does that impact the total number of values that can be encoded? If so, how many distinct values can now be encoded?

Almost all modern computers use a 2's complement representation for integers since the 2's complement addition operation is the same for both positive and negative numbers. In 2's complement notation, one negates a number by complementing each bit in its representation (i.e., changing 0's to 1's and vice versa) and adding 1. By convention, we write 2's complement integers with the most-significant bit (MSB) on the left and the least-significant bit (LSB) on the right. Also by convention, if the MSB is 1, the number is negative; otherwise it's non-negative. Use a 16-bit 2's complement representation to answer the following questions:

- c) What is the binary representation for 0? Answer:

- d) What is the binary representation for the most positive number that can be represented? Answer:

- e) What is the binary representation for the most negative number that can be represented? Answer:

- f) What is the decimal value for the most positive number? Answer:

- g) What is the decimal value for the most negative number? Answer:

- h) What is the result of negating the largest negative integer? Answer:

**Problem 3: Hexadecimal Representation (15 points)**

Since writing a long string of binary digits gets tedious, it's often convenient to use hexadecimal notation where a single digit in the range 0—9 or A—F is used to represent adjacent groups of 4 bits (starting from the left). Give the corresponding 8-digit hexadecimal encoding for each of the following numbers:

a) 5053 (decimal)

Answer:

b) -60 (decimal)

Answer:

c) 1011 1010 0001 0011 0011 0111 1110 1101 (binary)

Answer:

d) 1010 1111 1100 0001 0000 0101 0011 0101 (binary)

Answer:

e) -3 (decimal)

Answer:

**Problem 4: Binary Arithmetic (27 points)**

Calculate the following using 8-bit 2's complement arithmetic (which is just a fancy way of saying to do ordinary addition in base 2 keeping only 8 bits of your answer). Remember that subtraction can be performed by negating the second operand and then adding it to the first operand. Give your answers in binary.

a)  $58 + 29$

b)  $66 - 17$

Operand 1:	<input type="text"/>
Operand 2:	<input type="text"/>
Result:	<input type="text"/>

Operand 1:	<input type="text"/>
Negated Operand 2:	<input type="text"/>
Result:	<input type="text"/>

**Problem 4: Binary Arithmetic (continued)**

c)  $17 - 66$

d)  $127 + 127$

Operand 1:	<input type="text"/>
Negated Operand 2:	<input type="text"/>
Result:	<input type="text"/>

Operand 1:	<input type="text"/>
Operand 2:	<input type="text"/>
Result:	<input type="text"/>

e) Explain in what sense your answer for part d is correct.

**Problem 5: Fixed-Point Binary (16 points)**

Using a 16-bit fixed-point binary representation, where the leftmost 8 bits are the integer part (i.e., before the binary point), and the rightmost 8 bits are the fractional part, convert the first two decimal numbers below into binary, and the remaining two from binary into decimal.

a) 117.15625	Answer:
<input type="text"/>	

b) -8.25	Answer:
<input type="text"/>	

c) 0101 1110 . 1010 0000	Answer:
<input type="text"/>	

d) 1111 1111 . 1000 0000	Answer:
<input type="text"/>	